Goal Driven Enterprise Modelling: Bridging Pragmatic and Semantic Descriptions of Information Systems

Remigijus GUSTAS[1], *Janis BUBENKO jr.*, *Benkt WANGLER*

Swedish Institute for Systems Development
Electrum 212, S-164 40 Kista, Sweden
e-mail: {gustas, janis, benkt}@sisu.se

Abstract

Concepts, relationships and rules in the specification of requirements of information systems exist for some reason. They express the needs and rationale of requirements at pragmatic level. On the other hand, goals justify and explain the presence of requirement components. The problem here is to bridge the gap between the description of initial requirements and the precise specification of functional requirements of information systems. Explanation of goals in terms of static and dynamic constraints is of interest, because it allows the interpretation of goals as a driving force in the process of requirements modelling.

The objective of this paper is to describe some pragmatic dependencies using several types of semantic relationships in the context of goal driven information system modelling. The purpose for introducing the interaction between different classes of semantic and pragmatic dependencies is that developed specification of information system could be unambiguously interpreted among domain experts, and that it could be used for assessment of the quality of requirements.

1. Introduction

Requirements modelling is a critical phase in development of computerised information systems, because consequences of errors, made in the early stages, are difficult to trace in the future. Usually, the price paid for requirements reengineering activities is relatively large. A great variety of problems, such as incompleteness, contradictions, ambiguities, and overspecifications arise during analysis of several stakeholders viewpoints.

Ideally, wishes of potential users could be described as initial objectives which constitute a driving force for further conceptual design activities. This idea was raised in several papers [Bube93], [SBWJ94]. In general, two directions are suggested for bridging goals(informal statements in natural language) to static and dynamic (formal)constraints of the universe of discourse:

1) Refinement of goals in terms of states which are essentially defined as logical expressions [DLFi93]. The language of logic is powerful enough for definition of semantics of any goal, but logical representations are inconvenient and less comprehensive to human beings. The complexity and unreadability of logical

[1]The author is on leave fromInformation Systems Dept., Kaunas University of Technology, Lithuania;e-mail: remigijus.gustas@if.ktu.lt

formulas is the main reason why information system designers and analysts often do not like to use logical specifications [SoZa92]. Secondly, we can not expect big success of logical approaches in supporting of distributed requirements engineering methodologies.

2) Goals can be refined into business rules, which are defined on the basis of an ER [Chen76] kind of model [Temp92]. The abstraction level of primitives of semantic models [HuKi87], [PeMa88], [RBP*91] or some knowledge representation methods is much higher compared to logical formalisms. They have been developed to capture some meaning as well as representing the structure of data. A characteristic feature of rules, defined at the business modelling level[Gust94a], is that they abstract from any particular logical data representation aspect, while concentrating on epistemological [Brac79] aspects, which are common to all logical representations [Diet91]. Although business modelling level rules are more readable and understandable for human beings, there is no complete understanding of rationales behind goals.

Business rules are of interest because they capture semantics of goals. Goals are important in several aspects:

1) goals may motivate completeness of requirements, or may provide some help in detection of overspecifications, i.e. to discover requirements which have no motivation,

2) goals justify and explain the presence of requirements components, not necessarily comprehensible to users of information system, when they are separated from each other [DLFi93],

3) goals may be used to understand responsibilities of actors, i.e. they provide the basis for assigning actions to agents of the system,

4) structure of goals provide basic elements for detecting and resolving conflicts that arise from different viewpoints. Goals lead to integration of requirements which should support them [DLFi93].

The three dimensions of requirements engineering (understanding, representation, and agreement) [Pohl93]illustrates the three major aspects of the problem of goal modelling. On a specification dimension we are dealing with understanding of semantics of goal. The representation dimension has to support the meaning of goals by expressing them in some formal notation. Finally, the agreement dimension is based on understanding of goals from different viewpoints. It has to support trade-offs among several goals, i.e. a negotiation process of distributed requirements engineering. Enterprise models [Bube93], [F3Co94] have been developed to make a strict distinction between essential business oriented aspects of specification and logical data processing aspects of representation. Nevertheless, the fundamental problem of description of semantics of goals in terms of static and dynamic constraints of the information system still remains. The objective of this paper is to illustrate a possibility of description of some pragmatic dependencies in terms of semantic relationships [Stor93].

The paper is organised as follows: In section 2 we discuss briefly some of the notions most commonly appearing in goal models. The third section presents the framework of enterprise modelling. We will in section4 describe several properties of goals, and address some important issues of bridging goals to semantic representations. The example of interplay between pragmatic and semanticdependencies is provided by section 5. Some future research issues regimen in the concluding section.

2. Goals and Related Pragmatic Categories

Goals are usually understood as states of affairs or situations [BaPe83] that should be reached or at least striven for. A state in this case ice boats described as a set of entities (objects), their attributes and the set off relationships between entities. What is not included in a goal state is not part of the goal. A more general approach is taken by Sibley [Sibl93] in considering policy to deal with a set of states, that can either be *within* or *outside policy* . The latter approach could make use of state based models and transition diagrams, where an object can reach a new state in a certain way if the conditions of the objects allow the transition.

The notion of intent must be added to a state approach to policy analysis. It does not suffice to talk about the *possible* prior and next states surrounding a present state only, but *why* a certain path through the transition diagram is chosen must be accounted for. An important part of this is studying the ways in which different goals are related.

A goal hierarchy could be formed of interconnected goals on different levels of abstraction ranging from high-level business objectives to low-level operational goals or business rules. Fig.2.1 gives an overview of such a breakdown of goals. It thus describes, in the downwards direction, how the various high-level objectives are going to be achieved and in the upwards direction gives the reasons for theintroduction of various lower level goals and rules satisfying those on higher levels.

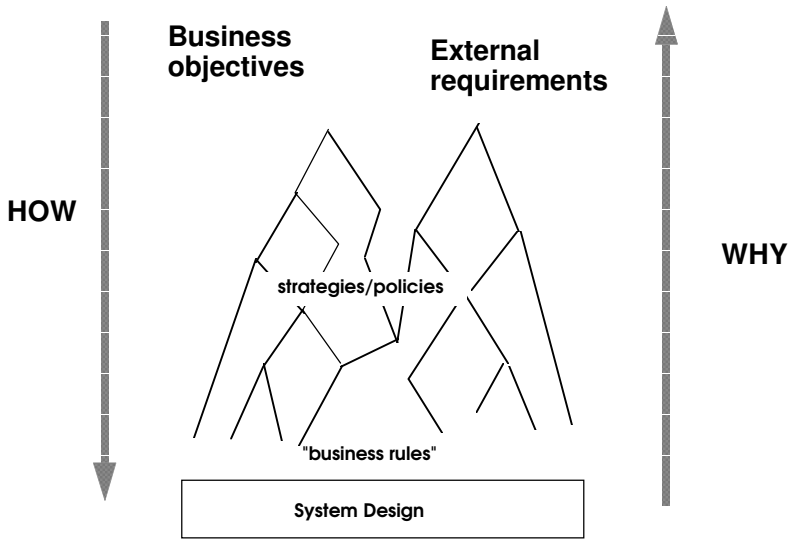


Fig.2.1. Goal breakdown.

Such a structure of various action-guiding or action-restricting notions(starting out with the most general) typically includes notions such as: objectives, strategies, policies, goals, requirements, rules, and constraints. In the following, we discuss briefly some of the notions most commonly appearing in goal models.

- **Notions of vague intentions** such as *goal, objective, mission, vision* all express wishes and desires concerning the properties of the business or the information system to be developed, albeit with different shades of meaning. For instance, to call some wish a requirement should normally indicate a stronger desire than to name it a goal. Sometimes the term objective is used as a synonym for goal with the extra semantics that it is now more strictly formulated and possibly measurable.

- **Notions of corporate control** such as *policy, strategy* and *business rule* are normally used for those intentions that express rules of control of the business. Policies and strategies express ideas about how to carry through the business, whereas business rules are usually introduced on looker levels in order to satisfyce higher level business goals. Those rules are built into and form the most important part of information systems. In order to acquire system maintainability these have to be easily traceable so that the system can be changed when the rules change.
- **Notions of restriction**, expressing various *constraints.* To name a particular requirement a constraint usually indicates that it has to be adhered to. It may for instance restrict the possible courses of action to take during the development of a system.
- **Notions expressing situations we want to avoid.** In practical goal modelling we often refer to these as problems. They may, for instance, express some recognised malady with the system. Some examples are 'customer complaints have doubled', 'book damage has increased by 25 % over the last year'.

  Problems are statements of understanding that convey the reasons why something has gone wrong. Problems may come about from analysis of observations or may be directly observed, for instance, "incorrect orders have increased and delivery deadlines are not always being met", is a problem statement resulting from certain customer complaints.

  Problems have to be analysed to discover the reason why something is a miss. Complex and wicked problems are often composed of several sub-problems with complex causal chains.
- **Notions expressing indicative situations** such as *causes* of problems, *prerequisites* or *opportunities*. Those are statements relating to what should be done to fix a problem or what may be carried out to improve an already satisfactory state of affairs. Opportunity statements should have some description of a state which may somehow help to remedy the problem, although they require further analysis before operational and technical decisions can be taken.
- **Actions** take you from one state of affairs to another, for instance from a problematic situation to one expressed by a goal statement.

At the lower level of a goal-hierarchy the degree of action (or possibly forbidden action) specified by the goals increases. As the rules and low-level business goals are concrete, often possible to operationalize and concern a specific matter, they are much easier to formalise than the more or less vague high-level business objectives and policy statements. Accordingly, goals are sometimes classified into strategic, tactical and operational goals.

3. Enterprise Modelling

The framework of Enterprise Modelling [F3Co94]includes a number of interrelated submodels. The semantic links established in these submodels are reflecting rationale and motivation for specific information system components [Bube93]. The enterprise model reflects design of the business. It shows the business entities, processes and how they interact. Information system model determines the data elements and functions that represent business entities and processes [SoZa92]. So, the enterprise model

deals with understanding perspective while information system model corresponds to representation perspective [Pohl93].

In the F3 approach, the following eight sub-models are suggested. The *objectives sub-model* describes, the "why", the long term intention or vision of the enterprise we are discussing or designing. The *concept sub-model* describes what "things" (phenomena) we are talking about, using a conceptual modelling technique. The *activities* of the enterprise as well as the *actors* involved in these activities, are described in corresponding sub-models of the requirements specification. These models may describe an existing situation or a future, desired situation. The requirements, *functional* as well as *non-functional*, we have on the supporting information system, are described in the *information system requirements* sub-model of the specification.

A requirements specification is not developed in a simple, straightforward fashion. Many intermediate, and sometimes alternative design versions may exist. In order to deal with this, we need models to describe different *configurations* of designs and specifications. Finally, we have to link the requirements descriptions to the formal design specifications of the information system, the *information systems sub-model*. This give us eight interrelated sub-models to be developed and maintained in requirements acquisition.

The reason to suggest these eight sub-models is the experience that there is a need and benefit to separate between particular concerns in requirements engineering. Each sub-model represents a particular concern and requires particular user and customer expertise. While there are certainly several ways of separating the concerns of requirements acquisition, it has been found that these sub-models separate the different concerns in a workable and natural way. The different sub-models are, however, not developed(populated) in a sequential, waterfall-like, fashion. It is true that the information system requirements and the information system sub-models are, typically, developed in the later phases of the requirements acquisition process. It is, however, not always the case that the process starts with developing an objectives model. Sometimes the situation of the modelling group may be such, that it is better to start developing some other model. For instance, if there is a general lack of common understanding of the important concepts of the enterprise (an observed situation of the modelling group), the group may well start with developing the concept sub-model in order to improve communication and understanding of what is being talked about.

The concept of "Enterprise" should here be interpreted in a very wide sense. It could actually mean the whole enterprise or some small part of it. More often than not, in developing information systems, the concept "enterprise" will denote a limited area of activity of the enterprise of which we are concerned. Examples of a limited area might be „office document management", "customer relations management", or „publicity management" of a company. An enterprise could also mean the process of developing a method and a computer based supporting tool-set. In this case the tool-set can be seen as the information system supporting the "method activities", which are consistent with some "method objectives". Depending on how familiar we are with the application domain, its conceptual structure, and the requirements for computer support, the need to perform an extensive, in-depth environment modelling may vary. For instance, in a limited technical situation, where the requirements and the conceptual model is well defined, the need for extensive enterprise modelling is limited.

The rest of this section describes the different sub-models in more detail. Our assumption is that developing a requirements specification, basically implies "population" of a number of meta-models. These meta-models describe the kinds of interrelated "documents" (or sub-documents) to be developed. By following a set of "good" meta-models, the risk to develop incomplete, and inconsistent specifications can be reduced. By

"good meta-models" we mean models which separate concerns in a way which improves clarity, as well as development work quality and productivity. Experience has shown [JaPo93], [Temp92],[Will93] that the use of an "objectives model driven approach" may lead to improved understanding of the problem realm for "end-users", decision makers, requirements holders and customers, as well as for developers. The economic value of improved communication and understanding for minimising re-work in later stages of the information systems development process , is well documented [CKIs88].

In F3 [F3Co94], it is proposed to structure a requirements specification in an number of "sub-models", also shown inFig.3.1.

- *The objectives sub-model*. In this part of a requirements specification we describe, in a structured way, the "why" component of a requirements document. *Goals* and *business rules* for a particular enterprise activity (or set of activities), existing, to be modified, or to be designed, will be stated, and their relationships analysed. Other component types of this sub-model are *problems*, *causes* (of problems), *opportunities*, and *development actions*. While goals and rules are the „important" components of this sub-model, the other types of components help and support the detection as well as formulation of goals and rules. They are hence considered as important text components in order to document and explain why certain rules and goals have been formulated. The objectives model is a graph with the above types of components as nodes, connected by relationships of type "motivates" or "influences". The motivates relationship is here seen as a refinement relationship (e.g. a goal is refined in a number of sub-goals). The influences relationship is a relationship indicating positive or negative influences between objectives model components. These relationships can also be given different levels of "strengths", e.g. low, medium or high.

- *The concept sub-model*. The concept sub-model is used to define the "ontology" of the "universe of discourse" that concerns us, i.e. the set of object types, relationships, and object properties of the application we are talking about. In this sub-model we also further refine business rules of the objectives model into static as well as dynamic rules for the concept sub-model states as well as for permissible state changes. For developing a functional information systems specification, the concept sub-model will define about what "things and relationships" information must be represented in the information system, and what should be the rules, implemented in the processes of the information system, according to which the information system should behave. The concept sub-model will, moreover, serve as a dictionary of user and customer defined concepts, and conceptual structures to be used to strictly express other parts of a requirements specification.

- *The actors sub-model*. This sub-model is used to discuss and define the set of actors of the studied activity (individuals, groups, job-roles/positions, organisational units, machines, etc.), and their inter-relationships, such as part-of, reports-to, etc. This model includes links to the other sub-models, e.g. who is the "stake-holder" of a particular goal, who is responsible for managing an activity according to a particular goal, who is the author of a non-functional requirement, etc.

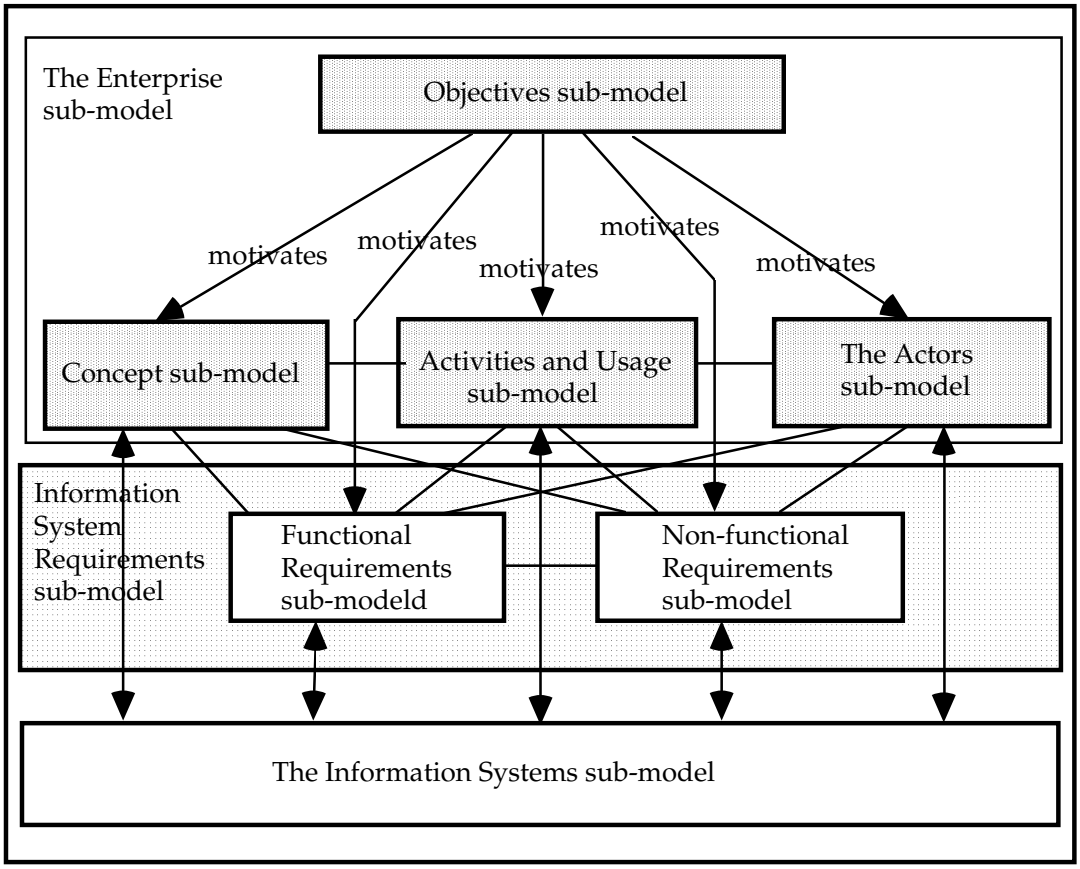


Fig.3.1. Interaction among the sub-models. Note that the objectives sub-model plays a dominant, motivational role, i.e. it describes the reasons for developing and including components in the other sub-models.

- *The activities and usage sub-model*. In this part of a requirements specification, the particular organisational activity (in a wide sense), existing, to be modified, or to be developed, is defined and described from the point of view of activities, tasks, processes, and the information and material flows between them. Describing, for instance, human - computer interaction is part of this sub-model. Clearly, components of this sub-model are motivated by components of the *objectives sub-model*, they are performed using or referring to components of the *concept sub-model*, and they are typically carried out by components (resources) of the *actors sub-model*.

Information system requirements, related to the above models, are described by the following two submodels.

- *The functional requirements sub-model*, This part of the requirements specification elaborates the specific objectives and requirements that are put on the information system to support the activities and objectives listed previously. Typically it will indicate needs for establishing objects (or relationships, attributes), deleting objects, modifying objects, and the need for checking, querying and browsing objects and relationships.
- *The non-functional (NF) requirements sub-model*. NF requirements are primarily related to the activity and usage model, and indirectly to the objectives sub-model, as activities normally are motivated by the objective model. NF requirements

concern components of the information systems sub-model, i.e. information subsystems. NF requirements can be of several different kinds, such as standards for interfacing other systems, restrictions concerning use of hardware and/or software, accuracy of information, timeliness of information, security, access-rights, economic limits of the development project, etc.

The information system is described conceptually by the *information systems sub-model*. By this we mean the complete, formal, functional specification of the information system. It is the task of the IS designer to develop IS designs and descriptions such that designated activities and processes in the activities and usage sub-model, as well as the functional and NF-requirements are supported (to a higher or lesser degree).

In order to manage the modelling process and its results, we need the *configuration sub-model.* This is a sub-model, orthogonal to the other sub-models. The main concepts of this sub-model are *version* and *release*. We need this dimension for the reason that systems, they may be organisational or computer based, continuously evolve and change. For instance, the concept sub-model, or parts thereof, will change at times. New versions of these sub-models will be developed, based on other earlier versions. On the other hand, we cannot disregard such historical information, as some actors may be still using it. Developing a new version of some sub-model will most likely also cause needs for change in the other sub-models. For instance, a new objectives sub-model description will certainly affect and require a re-consideration of all other sub-models. To manage "configurations" of descriptions is an important mission of a good requirements modelling facility.

What we have above is basically a set of interrelated descriptions or documents, each constructed with a predefined set of "permitted" types of components and component relationships. Depending on the *purpose* of the modelling exercise, we may, for a particular modelling job, need only a subset of these sub-models. Other sub-models may be, for a particular modelling purpose, superfluous and "empty". We may think of several other purposes for modelling. One such is the re-design of organisational activities, without assuming any supporting computer system to be developed. Another, quite typical purpose in practical situations, could be to reconsider the organisation's goals, objectives, mission, etc., and then study how an existing information system actually supports the new set of goals. However, in this paper we assume that a business activity is to be studied and that requirements for computer support for it are to be developed. This typically requires all sub-model types indicated above.

4. Pragmatic Dependencies of the Objectives Submodel and their Interaction with Semantic Representations

The ultimate goal of objectives modelling is to identify, to document and to analyse the business in terms of pragmatic[LLSo94] level entities such as goals, problems, causes of problems, opportunities, and pragmatic relationships. The objectives model addresses the 'why' perspective. It is motivating and driving the development of other enterprise submodels. So, one of the main difficulties, as everywhere in the area of requirements modelling, is unambiguous description of pragmatic level entities on the basis of modelling artefacts of concepts, actors and activities submodels. It can be done by defining pragmatic relationships in terms of static and dynamic constraints. In this chapter, the overview and description of some properties of pragmatic relationships are presented.

Objectives submodel components specify the pragmatic aspects [LLSo94] of an enterprise or an information system. Those aspects can be expressed on the basis of two specifications: the specification of actual situation and the specification of desired situation of the enterprise. The actual situation defines the static and behavioural constraints of actual specification, and the desired situation does it for the expected specification. The explanation of the dynamic relation between these two situations is depicted in Fig.4.1.

Fig.4.1. Relation between initial and desired specification

The development action describes activities that have to be performed in order to achieve a predefined set of goals. The activity forms a set of development tasks that have to be carried out [Bube93]. Let us consider the special case when the specification has to be developed for one particular goal of the enterprise. In this case, an actual situation may be specified by the problem statement, while the semantics of a desired situation can be refereed by a goal statement. Thereby, the development action takes us from the problematic to the expected situation, specified by the goal. This case is illustrated in Fig.4.2.

Fig.4.2. Relation between problem and goal

Another kind of dynamic dependencies are known as strategic dependencies[Yu94]. They exist as material, information or intention (control) flows between several actors of the universe of discourse. These dependencies belong to elements of the actors submodel. An actor is an active element that carries out a specification in order to achieve some goal. Each strategic dependency link between two actors specifies that one actor - recipient depends on the another actor - agent that may attain some goal by performing a specific action. A graphical notation of flow dependency is presented in Fig.4.3.

Fig. 4.3. The strategic dependency between actors

A strategic dependency is represented graphically by an arrow into and out of a process. The dependency could be used to describe the movement of chunks of information or material flow from one actor to another. Thus, the intentional flows represent behavioural aspects, whereas the situations denoted by problems and goals

represent changes of static structures. A graphical notation of a typical combined relation among agents, flow, action and two situations is presented in Fig.4.4.

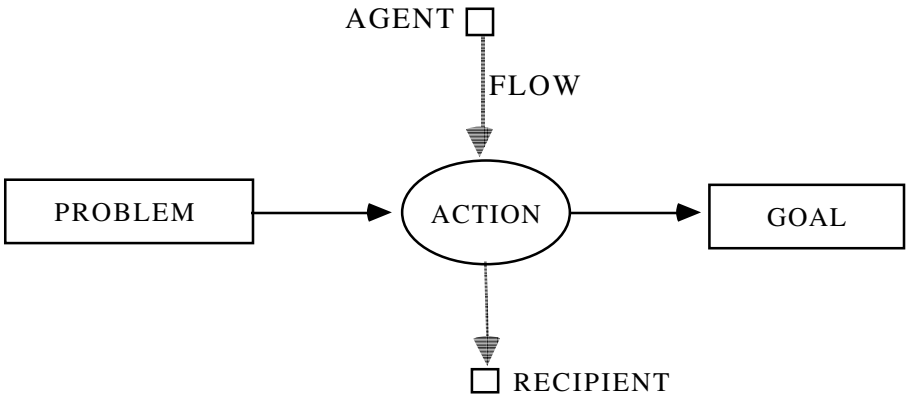


Fig. 4.4. Typical Dynamic Element

Here, the change of state is controlled by an external agent. The transition from the problem to the goal is performed by the action, if conditions stated by the actual situation are fulfilled and the intensional flow is sent from the agent. After the activation of the action, the conditions defined by the desired situation become fulfilled, the problematic conditions cease to exist and the flow is received by recipient. Such understanding of dynamics of actions is in correspondence with well-known modelling techniques such as Data Flow and State-Transition Diagrams [Your89]. Situations (states) can be specified on the basis of concept model dependencies [Gust94c].

The strategic dependency between actors describes goals and desires of an agent. The recipient, by depending on agent, is able to achieve new goals that the agent was not able to do separately. At the same time, the recipient becomes vulnerable [Yu94]. If agent fails to deliver intentional flow, the recipient would be unfavourably affected in his capability to achieve further goals. By matching the dependencies from agents to actors in the actors submodel, one can explore opportunities that are available to these actors. It should be noted that static dependencies among actors, such as aggregation and generalisation [SmSm77] which facilitate reasoning about opportunities and vulnerabilities of recipients, are defined by the concept submodel. The recipient becomes vulnerable to the failure of delivering the flow sent by the agent. In this case the recipient would be concerned about the viability of flow dependency.

The opposite of a goal is a problem. *A problem describes a situation which is not desirable*. This situation can be specified by set of static and dynamic constraints. *A specification of a problem* is defined as a semantic difference between a set of semantic constraints of the actual situation and a set of constraints of the desired situation. So, the semantics of a problem can be regarded as part of the specification of product at the actual situation. It is very important that *the problem can not exist without stating the goal*. If the designer has no predefined goal then the problem does not make sense. A goal describes a *desirable situation*. *Specification of a goal* is unambiguously defined as part of the semantic description of a desired situation which is deduced as a semantic difference between a set of constraints of the desired situation and the set of constraints of the actual situation. So, according to this understanding, the specification of goal is the opposite side of specification of problem. At the same time, the specification satisfyce [Simo84] the goal or the problem and is regarded as the *extension* of this pragmatic element.

Any two goals can be *contradictory*, i.e. one goal can influence negatively or hinder to the achievement of the other goal. It means that interpretations of the notion of a

goal and a problem are relative and dependent on actor objectives. The same description could be interpreted as a goal for one actor, and as a problem for another actor. To specify contradictions between several goals of agents, the negative influence dependency [F3Co94] is introduced. An influence from A to B (A ⇢ B) dependency means that situation denoted by B is affected by situation A. The negative influence pragmatic relationship(⇢) from one goal to another indicates that the first hinders to the achievement of the second. The positive influence dependency (⇢) would mean, that the achievement of some goal, automatically contributes to the achievement of the other.

The *pragmatic dependency of specialisation* of A by B is denoted A ⋖ B if and only if the specification of B is a subset of the specification of the pragmatic element A. The 'motivation' dependency [F3Co94] is usually used instead of specialisation to refine a goal. The 'cause' dependency can be used in the case of specialisation of a problem. So, the consistency of decomposition of complex pragmatic entities to simple ones can be verified as far as their structures are specified by semantic constraints. Graphical notations of types of pragmatic relationships are presented in Fig.4.5.

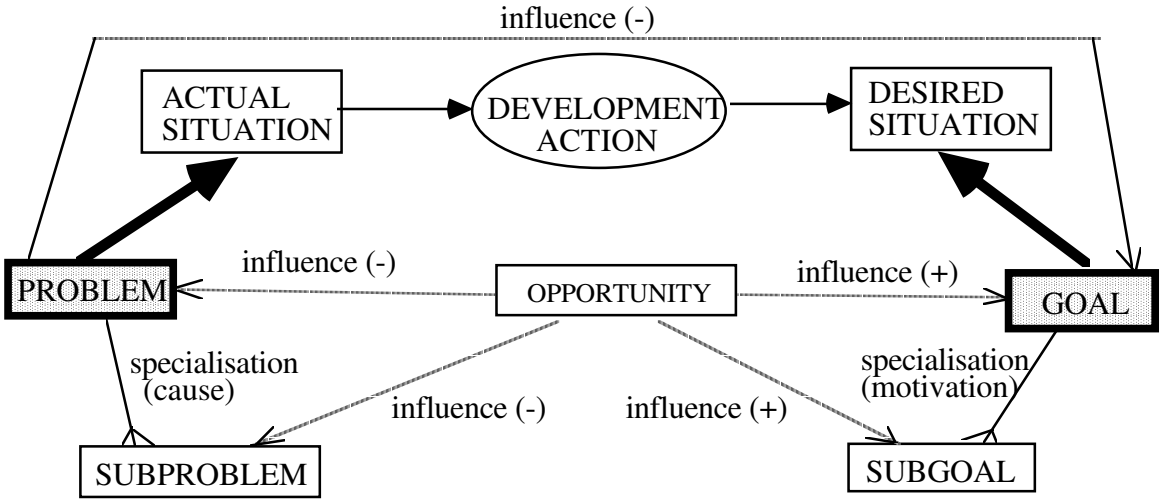


Fig.4.5. Types of pragmatic dependencies. Here ➡ is the inheritance (ISA [Brac79]) dependency.

Using the above described primitives, we can now explain notions of an opportunity and a cause of a problem introduced in the area of enterprise modelling [F3Co94]. *Opportunities* express situations that may be used to improve actual situations. It should be noted that the notion of an opportunity is relative and dependent on the methodology [F3Co94], [Will93]. Sometimes, opportunities are used for description achievable situations not regarded as final, i.e. situations that we may want to take advantage of. If so, they could be interpreted as intermediate goals. Opportunities can be specified using representations of the semantic level.

A cause of a problem can have different interpretations as well. One of the most simple definitions of a cause could be based on the notion of a situation which is the reason for a problem to exist. Causes are usually used for specification the explanations of problems and regarded as extensions of problems. Every specification of a subproblem is a cause for problem, and thereby, a subproblem causes a problem. The cause of the problem could be formed by taking the union of causes of subproblems. If the semantics of each cause can be expressed by exactly one constraint rule, then it shows higher quality of structuring in the phase of goal modelling.

The specialisation dependency relates more specific goals or problems with more general ones. If goal A is motivated by goal B then the extension of goal A (semantic

specification) constitute part of specification of goal B. It allows to define some useful operations on goals (problems), such as the operation of union and the operation of intersection of consistent goals. If *subgoals are conflicting* then the goal is *contradictory*. The goal A is *consistent* with the goal B if and only if A $\xrightarrow{+}$ B. Any contradictory goal has at least two subgoals which are not consistent .

Any two consistent goals G_2 and G_3 have their greater common specialisation denoted by intersection ($G_2 \sqcap G_3$) and their least common generalisation denoted by union ($G_2 \sqcup G_3$). A *union of goals* is defined as the union of their specifications. If any two goals G_2 and G_3 are specialisation of the goal G_1 (see Fig.4.6,a) then their union is the specialisation the same goal as well.

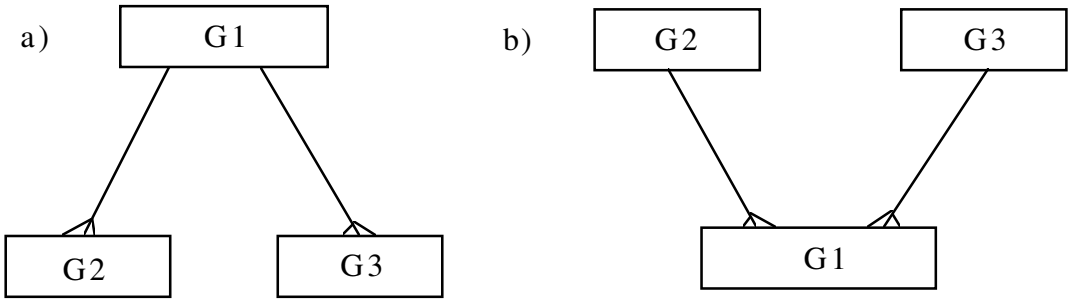


Fig.4.6. Operations with goals

Operation of union of goals can be characterised by the following properties:
1) $(G_2 \sqcup G_3) \rightarrowtail G_1$ if and only if $G_2 \rightarrowtail G_1$,$G_3 \rightarrowtail G_1$,
2) $G_2 \rightarrowtail (G_2 \sqcup G_3)$,
3) $G_3 \rightarrowtail (G_2 \sqcup G_3)$;

Intersection of goals is defined as intersection of extensions of goals. If any goal $G_1$is the specialisation two goals G_2 and G_3 (see Fig.4.6,b) then it is the specialisation of intersection of these goals. The operation of intersection of goals can be characterised by the following properties:
1) $G_1 \rightarrowtail (G_2 \sqcap G_3)$ if and only if $G_1 \rightarrowtail G_2$,$G_1 \rightarrowtail G_3$,
2) $(G_2 \sqcap G_3) \rightarrowtail G_2$,
3) $(G_2 \sqcap G_3) \rightarrowtail G_3$.

Union of any two contradictory goals and problems are equivalent to the empty element ($\perp$), i.e. $(A \sqcup B) \equiv \perp$. Two pragmatic elements A and B are *contradictory* if any specialisation of A ($X \rightarrowtail A$) influences negatively B($X \xrightarrow{-} B$).

If any two goals and problems are *independent* then the extension of intersection of them is equivalent to the empty element as well, i.e. $(A \sqcap B) \equiv \perp$. Specifications of independent pragmatic entities have no common semantic constraints.

One goal can be interpreted as a problem for another stakeholder. Furthermore, the pragmatic links can be defined by several domain experts in parallel and, therefore, be in conflict. The goal A is *inconsistent* with the goal B if and only if A $\xrightarrow{-}$ B and A $\xrightarrow{+}$ B. It should be noted that the negative (positive) influence pragmatic dependency can be derived [Gust95]according to the following inference rules:
1) if A $\xrightarrow{+}$ B , B $\xrightarrow{-}$ C then A $\xrightarrow{-}$ C ,
2) if A $\xrightarrow{-}$ B , B $\xrightarrow{-}$ C then A $\xrightarrow{+}$ C ,
3) if A $\xrightarrow{+}$ B , B $\xrightarrow{+}$ C then A $\xrightarrow{+}$ C ,
4) if A $\rightarrowtail$ B then A $\xrightarrow{+}$ B .

Using set of pragmatic relationships it is possible to infer some other valid (not specified) dependencies of the objectives model. The dependencies are derived according to the following inference rules:

1) if A $\succ\!\!-$ B ,C $\dashrightarrow$ A then C $\dashrightarrow$ B ,

2) A $\succ\!\!-$ B , B $\succ\!\!-$ A if and only if A $\succ\!\!\prec$ B ,

3) if A $\succ\!\!-$ B , B $\succ\!\!-$ C then A $\succ\!\!-$ C ;

Here $\succ\!\!\prec$ is the mutual specialisation dependency. The mutual specialisation dependency is useful for decomposition of goals, i.e. if (($G_2 \sqcup G_3$) $\succ\!\!\prec$ G_1) the G_2 and $G_3$is decomposition of G_1, i.e. components G_2 and G_3 are sufficient for refinement of goal G_1.

5. Example of Interplay between Semantic and Pragmatic Relations

In this chapter, the library example is presented to illustrate an interaction among semantic and pragmatic constraints. The objectives and problems of the library users can be covered by one general goal which is entitled 'Well-functioning Library'. One of the features of the library, to be well-functioning from the point of view of the borrower, is 'user-friendly borrowing'. In order to satisfy the objective of the user-friendly borrowing, the borrower would like to have high availability of books, be able to borrow all kinds of books, and to have a long loan duration, i.e. (User-friendly Borrowing) $\succ\!\!\prec$ ((High Availability of Books) $\sqcup$ (Long Loan Duration) $\sqcup$ (Loan of All kinds of Books)) . It means that the library user has three reasons for the problem 'Borrowing isn't User-friendly', i.e. (Borrowing is not User-friendly) $\succ\!\!\prec$ ((Low Availability of Books) $\sqcup$ (Short Loan Duration) $\sqcup$ (Impossible to Loan All Books)) .The objective model dependencies of the borrower are depicted in Fig.5.1.

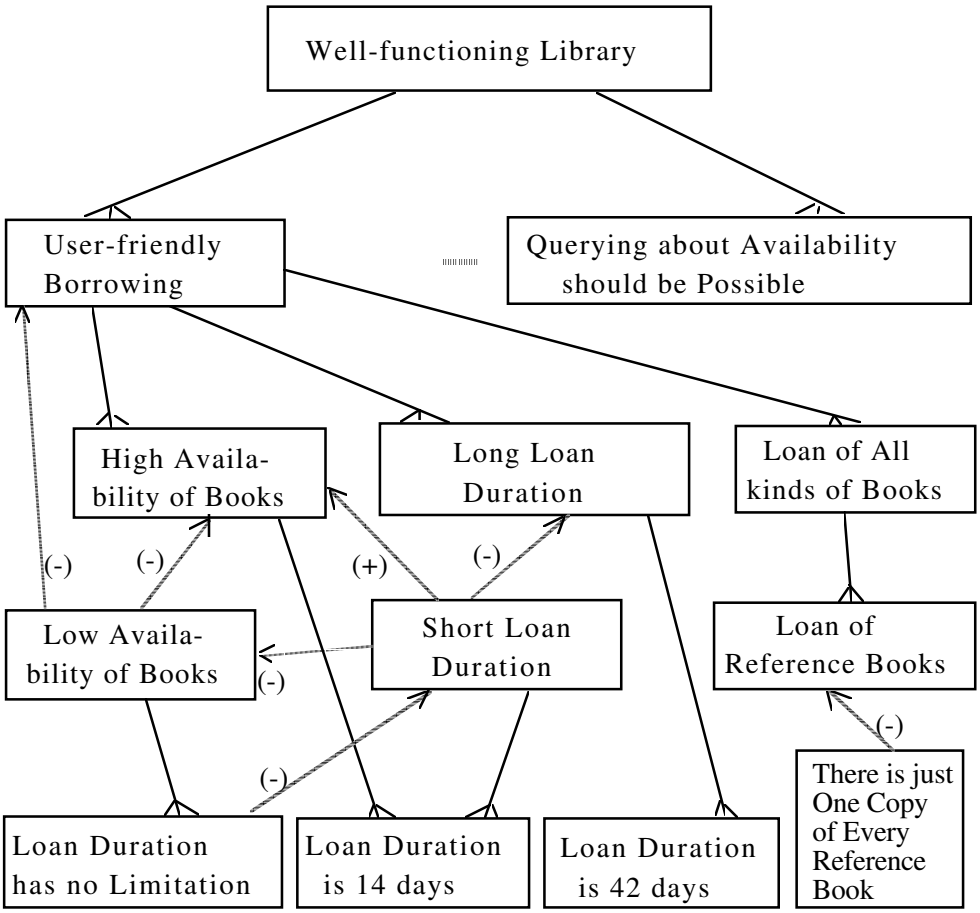


Fig.5.1. Objectives and Problems of the Borrower

According to the presented pragmatic dependencies of the borrower, is not difficult to find out the semantic difference between the situations denoted by problems and goals of the borrower. Lets consider the problem 'Low Availability of Books' which is the reason for the objective of borrower 'High Availability of Books' to exist. The reason, why the borrower wants to have 'long loan duration' is that the current loan duration is too short. It is caused by the requirement 'Loan Duration is 14 days', i.e. the librarian allows loaning just for 14 days. The problem in this case could be specified as follows: **Loan. Duration = 14** . In contrast to the librarian, the borrower would like to keep a book at least for 42 days, that is: **Loan. Duration = 42** . Notice that the subgoal 'High Availability of Books' was introduced by the problem of 'Low Availability of Books' before. It was caused by the reason that 'Loan Duration had no Limitation'.

Some of the static constraints among elements of the concept model can be described in terms of an ER diagram. The diagrams are presented from the viewpoints of two actors: the librarian and the borrower. Static requirements from the viewpoint of the librarian are depicted in Fig.5.2.

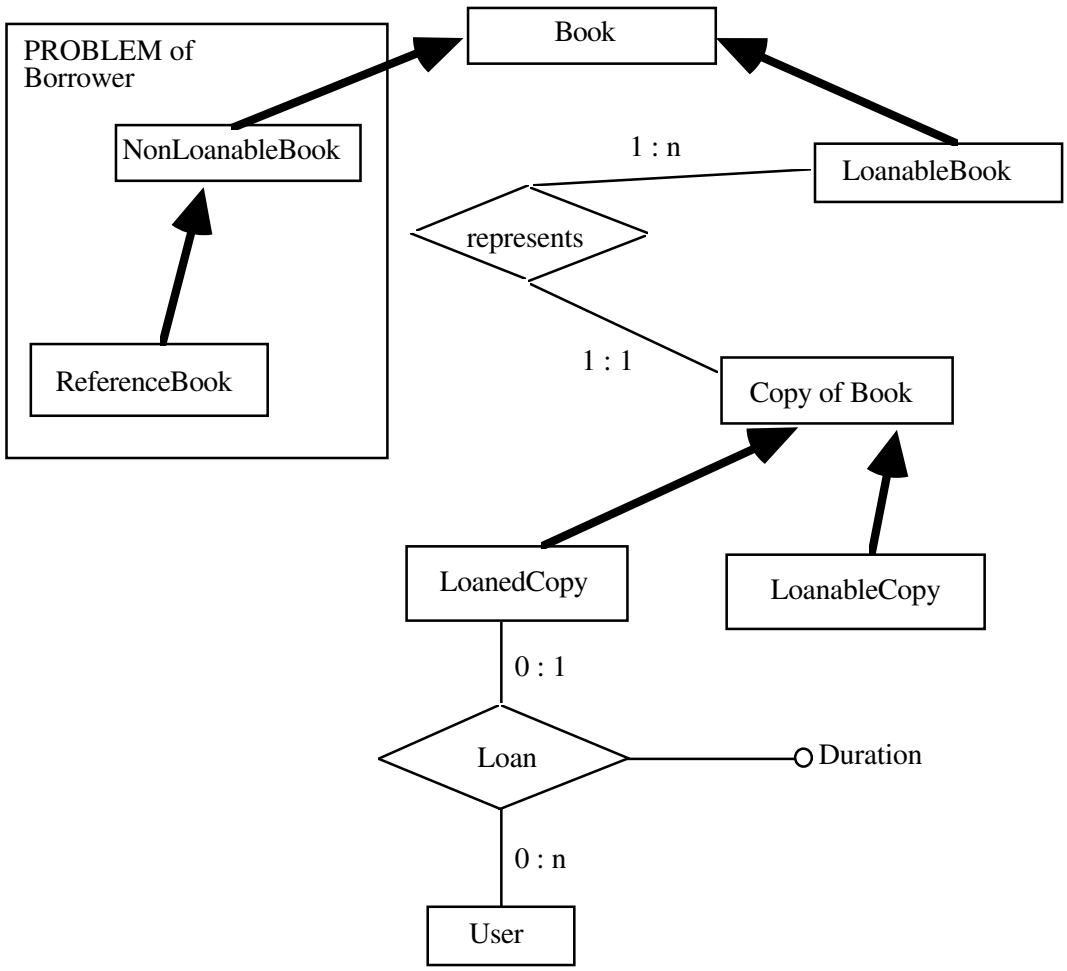


Fig.5.2. Static Requirements of the Librarian

In contrast to the borrower, the librarian wants ReferenceBooks to be always present in the library, i.e.

ReferenceBook ⟶ NonLoanableBook .

According to the presented requirements of the borrower (Fig.5.3), we find that Reference Books should be loanable, i.e.

ReferenceBook ⟶ LoanableBook .

This minor difference in the specification of static requirements of librarian gives rise to one of the causes of problems in the requirements of the borrower. Static requirements from the viewpoint of the borrower are depicted in Fig.5.3.

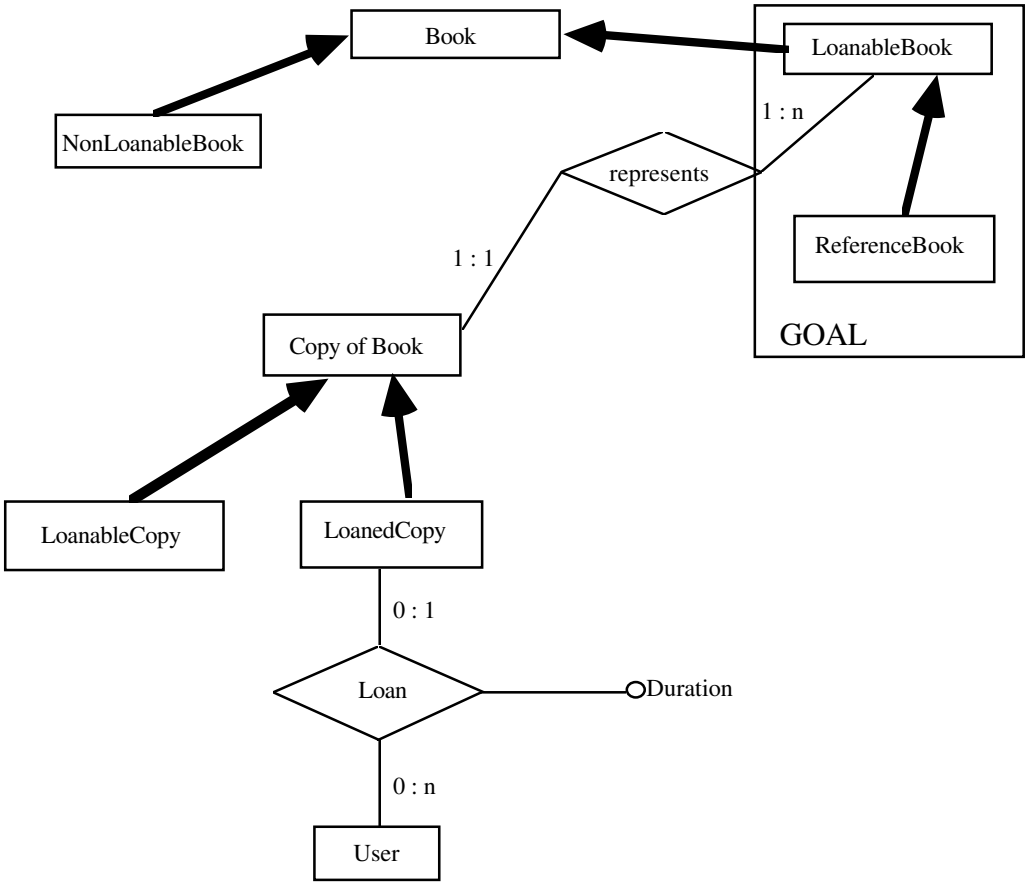


Fig.5.3. Static Requirements of the Borrower

According to the presented requirements it is not so difficult to understand the semantic difference between the expected situation denoted by the goal 'Loaning of All Books' and the problematic situation 'Impossible Loaning of All Books' in terms of semantic subschemes of two different actors - the borrower and the librarian. If the librarian wants reference books to be always present at the library (see Fig.5.2), while borrower wants them to be loanable (see Fig.5.3), then the meaning of the goal and the problem of the borrower could be defined as follows:

Problem: Impossible Loaning of All Books.

Specification of the cause (extension) of the **Problem**:

ReferenceBook ⟶ NonLoanableBook .

Goal: Loaning of All Books.

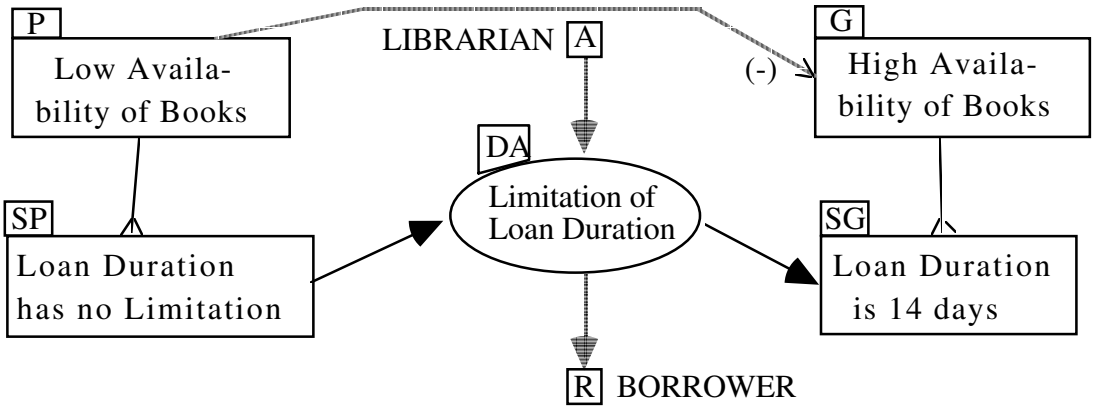
Specification of the meaning of the **Goal**:

ReferenceBook ⟶ LoanableBook .

The important point here is that the specification of the meaning of this particular goal and problem of the borrower can be represented as a semantic difference between two diagrams. The reason, why the user wants 'Loaning of All Books' is that he would like reference books to be loanable. 'Loaning of All Books' is not allowed because of the reason that ‘There is just one copy of every reference book'. This pragmatic entity can be interpreted as the problem of the librarian as well.

The goal 'High Availability of Books' was introduced by the librarian before. It was caused by the reason that 'Loan Duration had no Limitation' contributed to the problem 'Low Availability of Books’. The development action in Fig.5.4 'Limitation of

Loan Duration' takes the borrower from the situation 'Loan Duration has no Limitation' to the situation 'Loan Duration is 14 Days'. Notice that, from the perspective of another goal, the subgoal of the borrower (in Fig.5.4)'Loan Duration is 14 days' could be interpreted as the subproblem. For example, the situation 'Loan Duration is 14 days' is interpreted as the cause of problem in the context of the goal 'Long Loan Duration' (Fig.5.5).



Here: P - problem, SP - subproblem, G - goal, SG -subgoal, DA - action, A - agent, R recipient.

Fig.5.4. Problems and the development action in the context of the goal 'High Availability of Books'

The strategic dependency was directed from the librarian to the borrower. It specifies how goals and desires are dependent on the development action of the librarian 'Limitation of Loan Duration'. It can happen that the borrower, by depending on the librarian, is able to achieve his goal 'High Availability of Books' that was impossible to achieve independently. At the same time, the borrower becomes vulnerable [Yu94]. If the librarian fails to deliver intentional flow, i.e. if the development action 'Limitation of Loan Duration' is not successful, then the borrower would be unfavourably affected in its capability to achieve his goal 'High Availability of Books'. On the other hand, the borrower could become vulnerable in the context of the contradictory goal. Let us consider the goal of the borrower (Fig.5.5)'Long Loan Duration'. The goal 'Long Loan Duration' is in conflict with the cause 'Loan Duration is 14 days'. In Fig.5.4., this cause could be interpreted as the subgoal of the borrower. So, by accepting the decision of the librarian introduced by the development action 'Limitation of Loan Duration', the borrower would be unfavourably affected in its capability to achieve his goal 'Long Loan Duration'.

According to the requirements of the borrower, we can now find out extensions of some complex derived goals and problems. If we know that the user wants to borrow books for 42 days, in contrast to the librarian who allows loaning just for 14 days, the specification of some complex pragmatic entities will be as follows (see Fig.5.1):

- Goal**: (High Availability of Books)** $\sqcap$ **(Short Loan Duration)**; Specification: **Loan. Duration = 14** .
- Goal**: (Loan of All kinds of All Books)** $\sqcup$ **(Long Loan Duration)** ;Specification: **Loan. Duration = 42 , ReferenceBook** $\rightarrow$ **LoanableBook** .

If any two entities of the objectives submodel are contradictory then their union is empty (see chapter 4). For example:

- Goal**: (High Availability of Books)** $\sqcup$ **(Long Loan Duration)** $\equiv \perp$.

If any two entities of the objectives submodel are independent then their intersection is empty (see chapter 4). For example:

- Goal**: (Loan of All kinds of Books)** $\sqcap$ **(Long Loan Duration)** $\equiv \perp$.

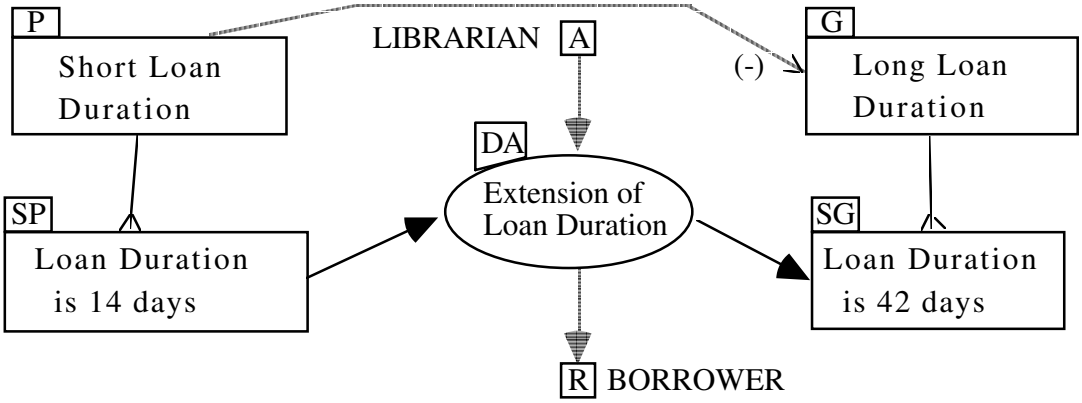


Fig.5.5. Problems and the development action in the context of the goal 'Long Loan Duration'

Real enterprise modelling situations, of course, are more complicated in comparison with the presented example. Nevertheless, the described set of dependencies can be regarded as a uniform basis of goal modelling. It provides capabilities to discover contradictions among goals. The set of inference rules of dependencies, presented in this paper, allows to the domain expert to discover additional links between several pragmatic elements. So, derived set of dependencies can be regarded as the basis for investigation of inconsistencies (conflicting links) among goals. It should be noted, that type of every pragmatic entity in the objectives submodel is the same, despite of types of links the elements are connected. Furthermore, the interpretation of a specific pragmatic entity as a problem, a goal, a reason of a problem or an opportunity is relative, and it could be derived according to the specified dependencies in the context of the specific goal.

6. Concluding Remarks

The starting point for requirements modelling research are initial requirements statements. These informal statements express wishes of various stakeholders about what the system should do or not do. Initial requirements are usually interpreted as objectives, opportunities, problems, and presented in the form of natural language text. These pragmatic representations are often ambiguous, incomplete and inconsistent. It makes the process of requirements modelling very complex. To simplify this process, the designer should be able to deal with several dimensions of requirements separately. These dimensions describe different aspects of requirements and correspond to syntactic, semantic and pragmatic level [LLS94] of business modelling. The mixture of modelling constructs from these three different levels in the same specification language, makes the application development process very complex, because the same set of requirements can be understood in three different perspectives [F3Co94].So, one of the main objectives of this study is separation, identification and bridging of modelling primitives of semantic and pragmatic levels. The graphical interpretation of three levels in information system requirements modelling is presented in Fig.6.1.

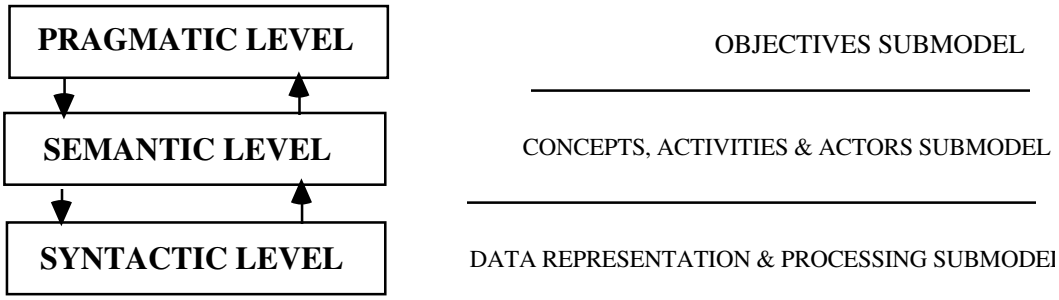


Fig.6.1. Three levels of requirements modelling

The syntactical level is closer to the implementation perspective. Modelling primitives of the syntactic level are well-understood. For example, relational database theory is atypical representative of this level. Semantic or object oriented datamodels are trying to capture modelling primitives of both semantical and syntactical levels. Perhaps, it could be a reason or good explanation of the fact, why we nowadays have no theoretical grounds in the field of requirements modelling. Data flowdiagrams [Yourd89] and ER kind [Chen76] of models [RBPEL91] are dealing with these two perspectives as well. To our opinion, it is to complicatedto deal with data representation, data processing, semantic and pragmatic aspects at the same time. This is similar to the procedure of direct explaining business goals in terms of program modules and file structures.

Nowadays, a great variety of commercially marketed information system development methods and CASE tools exist. However, these products deal with semantic and syntactic level, i.e. address mainly the middle and late stages of the systems development life-cycle. Practically none of them address the early, business objectives analysis, and the problem moving from the vague to the formal specification of requirements. Existing methods are not adequate for explicit capturing, and representing business objectives and problems, to be subsequently used to drive the later information system development phases [Bube95]. Links between business models and information system specifications are not maintained. This makes it impossible to reason about changes required in several information system 'worlds', as a consequence of changes of the enterprise objectives. One of the possible sources of complexity in the case of goal modelling is the inflexibility of the semantic level representations. Traditionally enterprise modelling techniques are based on the entity notations provided by several links [F3_94], [Tempora92]. Links are established to capture semantic detail about the dependencies among resources, activities, actors, etc. and other enterprise modelling elements. The problem here is that all these notations are not provided by rules of reasoning, capabilities of integratability of representations defined in context [JaPo93] of different goals. Furthermore, the theoretical foundations of the existing notations are too weak to establish such capabilities on the basis of the existing artefacts.

In this paper we have presented an approach for bridging pragmatic representations to semantic relationships [Stor93]established in different classes of models. The idea was illustrated using the framework of enterprise modelling which has a large scope and includes several modelling capabilities. Enterprise submodels are based each on a number of dependencies and relationships of different types. They represent different 'worlds' of requirements and addresses the 'why', the 'how', the 'who' and the 'what' perspectives [SoZa92] in information system development. Additionally, the use of conceptual dependencies allows interpretation of the 'where' and the 'when' perspectives as well. It should be noted, that we are not proposing a new model for specification of

requirements. We are discussing, how the specific set of pragmatic dependencies could be mapped to several semantic representations. Of course, the question of identification of a minimal set of pragmatic dependencies is still open.

Finally, this paper outlined general principle of integration of dependencies in different enterprise submodels. Development of a computer aided requirements engineering methodology requires a certain degree of formalisation and investigation of the presented models. Consequently, the further research work will be carried out to identify a minimal set of semantic and pragmatic dependencies for the purposes of enterprise modelling. A sound and complete theoretical basis could contribute for building foundations of requirements engineering. Dependencies constitute a uniform basis of the process of requirements modelling. But it has not yet dealt with principles of normalisation [RoCau92], and inference rules of semantic and pragmatic dependencies. None of models provides capabilities to prove such desirable properties as minimality and inconsistency of a set of specified links among entities of the enterprise submodels. The inference rules of pragmatic dependencies, presented in this paper, and inference rules of concept dependencies [Gust94b] can be regarded as a basis for investigation of redundancies and conflicts among several enterprise submodels links.

The ultimate objective of this paper is to introduce principles for modelling of pragmatic dependencies on a basis of different classes of semantic representations. The uniform understanding of basic modelling primitives could contribute to unambiguous definition of several pragmatic links in the context of enterprise modelling. The purpose for introducing such homogeneous understanding in the phase of goal modelling is that, finally, a developed specification could be unambiguously interpreted among domain experts and, thereby, it could be used for co-operative work between individuals as well as groups, where co-ordination is vital.

Acknowledgements

This work was partially funded by ESPRIT Basic Research Action 6353entitled NATURE which is concerned with Novel Approaches to Theories Underlying Requirements Engineering. NATURE partners are RWTH Aachen(Germany), University of Paris I (France), City University (London, UK), FORTH Research Institute (Crete, Greece) and Swedish Institute for Systems Development (Kista, Sweden). The participation of SISU in this project is sponsored by NUTEK, Swedish National Board for Industrial and Technical Developments.

7. References

[BaPe83]
Barwise, J. and J. Perry, Situations and Attitudes, The MIT Press, Cambridge, Mass, 1983.

[Brac79]
R Brachman. On the Epistemological Status of Semantic Networks. Associative Networks, Academic Press, pp. 3-50, 1979.

[Bube93]
J A Bubenko. Extending the Scope of Information Modelling. Fourth International Workshop on the Deductive Approach to Information Systems and Databases, Polytechnical University of Catalonia, pp. 73-97, 1993.

[Bube95]
J A Bubenko. Challenges in Requirements Engineering. Proc of the Second International Symposium on Requirements Engineering, York, England, March 27-29, 1995.

[Chen76]
P Chen. The Entity-Relationship Model: Towards a unified view of data. ACM Transactions on database systems 1, 1, 1976, pp. 9-36.

[CKIs88]
B Curtis, H Krasner, N Iscoe. A Field Study of the Software Design Process for Large Systems. CACM, 31(11), 1988.

[Diet91]
J L G Dietz. The Essential System Model. Third Int. Conf. on Advanced Information System Engineering CAiSE'91, Springer-Verlag, pp.441-460, 1991.

[DLFi93]
A Dardenne, A van Lamsweerde, S Fickas. Goal Directed Requirements Acquisition. Science of Computer Programming, 20, pp. 3-50, 1993.

[F3Co94]
F3 Consortium. 'F3Reference Manual (Esprit III Project 6612)', SISU, Kista, Sweden, 1994.

[Gust94a]
R Gustas. From Conceptual to Business Modeling. Proc. of Baltic Workshop on National Infrastructure Databases - Baltic DB'94, Vilnius, Lithuania, 17-20 May, pp. 218-229, 1994.

[Gust94b]
R Gustas. Towards Understanding and Formal Definition of Conceptual Constraints. The Fourth European - Japanese seminar on Information Modeling and Knowledge Bases, May 31 - June 03, 1994, Kista, IOS Press, pp.381-399.

[Gust94c]
R Gustas. Towards Foundations of Database Schema Independent Modelling of Concepts. Proc. of the Second International East-West Database Workshop, September 25-28, 1994, Klagenfurt, Springer-Verlag, pp. 275 - 288. .

[Gust95]
R Gustas. On Related Pragmatic Categories and Dependencies within Enterprise modelling. In Proc. of the Second Scandinavian Research Seminar on Information and Decision Networks, May 11 - 13, Vaxjo, Sweden, 1995.

[JaPo93]
M Jarke, K Pohl. Establishing Visions in Context: Toward a Model of Requirement Engineering.14th Int. Conf. on Information Systems, Orlando, USA, 1993.

[HuKi87]
R Hull, R King. Semantic Database Modelling: Survey, Applications, and Research Issues. ACM Computing Surveys, Vol.19, No.3, pp.201-260,1987.

[LLSo94]
O I Lindland, G Sindre, A Solvberg. Understanding of quality in conceptual modelling. IEEE Software,pp.42-49, 1994.

[PeMa88]
J Peckham, F Maryansky. Semantic Data Models. ACM Computing Surveys, Vol.20, No.3, pp.153-189,1988.

[Pohl93]
K Pohl. The three Dimensions of Requirements Engineering. Int. Conf. on Advanced Information System Engineering CAiSE'93, Springer-Verlag, 1993.

[RBP*91]
J Rumbaugh, M Blaha, W Premerlani, F Eddy, W Lorensen. Object-Oriented Modelling and Design, Prentice Hall Int., 1991.

[RoCa92]
C Rolland, C Cauvet. Trends and Perspectives in Conceptual Modelling. Conceptual Modeling, Databases and Case: An Integrated view of Information Systems Development, John Wiley & Sons, pp.27-48, 1992.

[SBWJ94]
A Sutcliffe, J A Bubenko jr., B Wangler, S Jacobs. Goal Driven Requirements Engineering (working paper), SISU, Sweden.

[Sibl93]
E H Sibley. Experiments in Organisational Policy Representation: Results to Date. IEEE Systems Man and Cybernetics Conference, Le Touquet, France, 1993.

[Simo84]
H A Simon. The Sciences of the Artificial. MIT Press, 1984.

[SmSm77]
J M Smith, D C P Smith. Data Base Abstractions: Aggregation and Generalization. ACM Transactions on Database Systems, Vol.2, No2, pp. 105-133, 1977.

[SoZa92]
J F Sowa, J A Zachman. Extending and Formalizing the Framework for Information Systems Architecture. IBM Systems Journal, 31(3), pp. 590 - 616, 1992.

[Stor93]
V C Storey. Understanding Semantic Relationships. VLDB Journal, Vol.2, pp.455-487, 1993.

[Temp92]
TEMPORA Consortium.'Tempora (P2469) Project Manual', BIM Information Technology, Belgium,1992.

[Yourd89]
E Yourdon. Modern Structured Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1989.

[Yu94]
E S K Yu. Modelling Strategic Relationships for Process Reengineering. PhD theses, Dept. Of Computer Sciences, University of Toronto, Canada, 1994.

[Will93]
H Willars. TRIAD Modelleringshandbok: Business Modelling Overview, SISU, 1993.